Volumetric *k*-means clustering for 3D SEISMIC FACIES ANALYSIS – PROGRAM kmeans3d

Overview

K-means (MacQueen, 1967) is perhaps the simplest clustering algorithm and is widely available in commercial interpretation software packages. **kmeans3d** is a volumetric seismic classification application that takes multiple seismic attributes as input, and generates a facies volume using *k*-means clustering. In **kmeans3d**, a user needs to specify the number of facies (clusters) to be generated, which is the *k* value in the *k*-means algorithm. Because the size of seismic data, in this implementation the software only takes a subset of data, extracted using a user specified decimation rate, to build a classifier, then apply this classifier to the whole 3D volume.

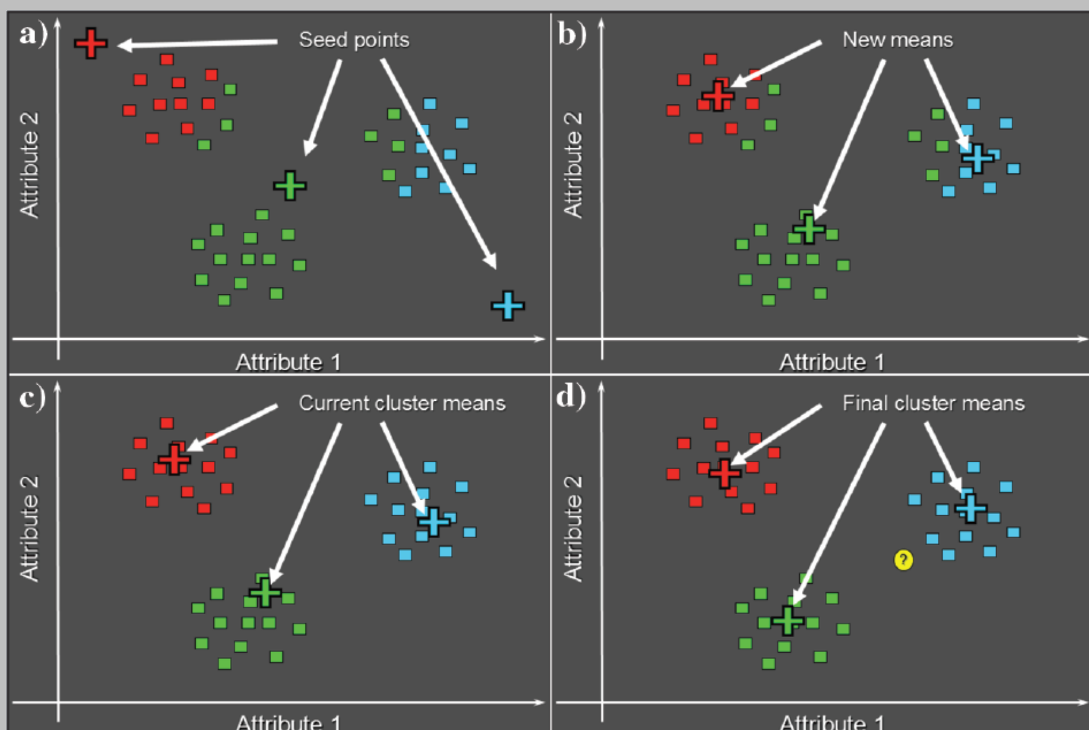
K-means is fast and easy to implement. Unfortunately, the clustering has no structure such that there is no relationship between the cluster numbering and the proximity of one cluster to another. This lack of organization can result in similar facies appearing in totally different colors, confusing the interpretation. Therefore, special care are needed when interpreting facies maps generated by *k*-means.

Volumetric Classification: Program **kmeans3d**

Theory

The workflow of computing *k*-means in **kmeans3d** is described here:

1. Decimate the input seismic attributes and store data in attribute space, i.e. each dimension is a seismic attribute.
2. Compute the eigenvalues, eigenvectors, and covariance matrix of input data in the attribute space, and use these to normalize data (same as z-score, which is for each data vector, measuring how many standard deviations away from the mean).
3. Initialize *k* centroids (mean of each cluster) uniformly in the normalized attribute space.
4. Perform *k*-means iteratively until change in centroids is smaller than a threshold value, or the algorithm reaches a user defined maximum number of iteration. The *k*-means updating process is shown in the figure below.
5. Apply the classifier to the whole dataset, assign each data vector to the cluster of the nearest centroid.

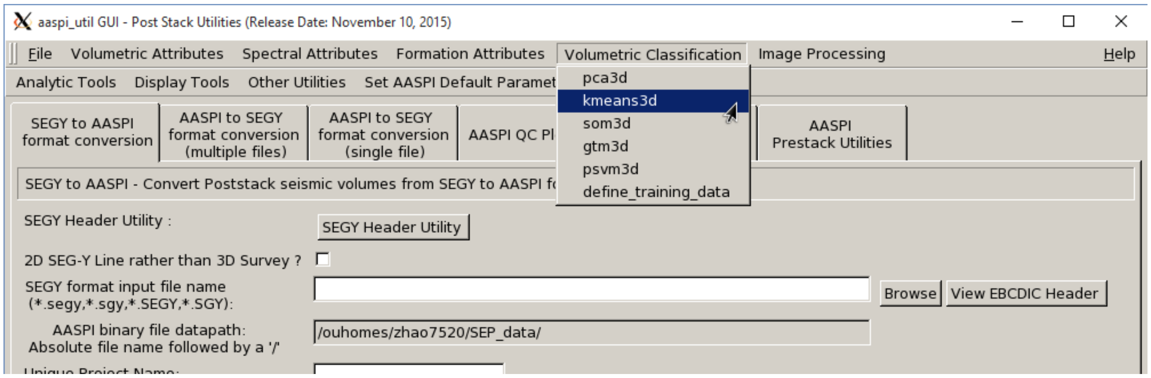


Cartoon illustration of a k-means classification of three clusters in a 2D attribute space. **(a)** Select three random or equally spaced, but distinct, seed points, which serve as the initial estimate of the vector means of each cluster. Next, compute the Mahalanobis distance (defined below) between each data vector and each cluster mean. Then, color code or otherwise label each data vector to belong to the cluster that has the smallest Mahalanobis distance. **(b)** Recompute the means of each cluster from the previously defined data vectors. **(c)** Recalculate the Mahalanobis distance from each vector to the new cluster means. Assign each vector to the cluster that has the smallest distance. **(d)** The process continues until the changes in means converge to their final locations. If we now add a new (yellow) point, we use a Bayesian classifier to determine into which cluster it falls (figure courtesy S. Pickford).

The Mahalanobis distance, r_{jq}, of the j^{th} sample from the q^{th} cluster center, θ_q, is defined as

$$r_{jq}^2 = \sum_{n=1}^{N} \sum_{m=1}^{N} (a_{jn} - \theta_{nq}) C_{nm}^{-1} (a_{jm} - \theta_{mq}),$$

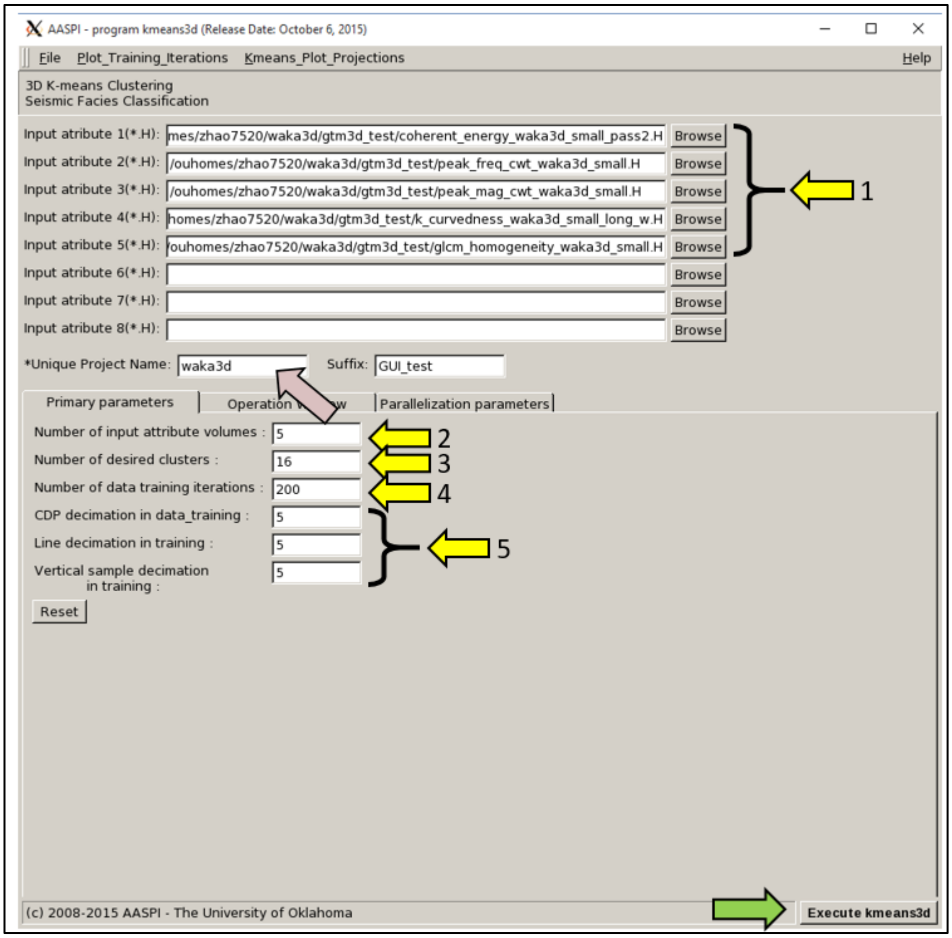
where the inversion of the covariance matrix, **C**, takes place prior to extracting the mn^{th} element.



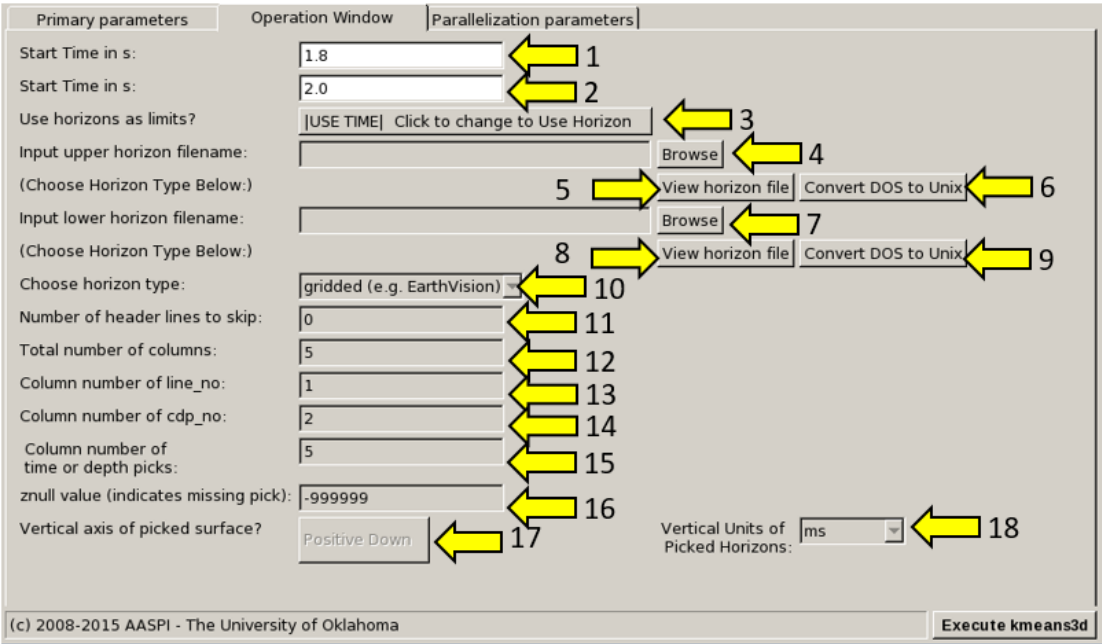
This Program **kmeans3d** is launched from the *Volumetric Classification* in the main **aaspi_util** GUI.

Computing kmeans3d module

Setting the primary parameters is the first step of analysis. Use the browser on the first eight lines to choose the input seismic data file (*Arrow 1*). It is not mandatory to take in eight inputs. The number of inputs can vary from two – eight. The input attributes that one considers for facies analysis will vary according to the specific applications. For identifying the depositional facies variation the volumetric attributes such as dip magnitude, coherency, GLCM attributes, spectral magnitude, coherent energy can be considered as input. For characterizing geo-mechanical variation in shale plays one should consider different volumes that helps in identifying the rock physics such as inversion volumes, lambda-rho, mu-rho, intercept or gradient AVO volumes, etc. Specify the number of input attributes in the field labeled “Number of attributes to use” (*Arrow 2*). This value will be updated automatically when a file is selected. Do not forget to give a “Unique Project Name”. A z-score algorithm is used to normalize the input files. The number of desired cluster is the *k* value in the *k*-means algorithm (*Arrow 3*). Empirically, we want a *k* value relatively small. Put the maximum *number of data training iterations* (*Arrow 4*). Select the decimation rate of input data used for training (*Arrow 5*).



Then a user need to define the operation window in the **Operation Window** tab shown below.


Primary parameters
Operation Window
Parallelization parameters
Start Time in s:
1.8
Start Time in s:
2.0
Use horizons as limits?
|USE TIME| Click to change to Use Horizon
Input upper horizon filename:
Browse
(Choose Horizon Type Below:)
View horizon file
Convert DOS to Unix
Input lower horizon filename:
Browse
(Choose Horizon Type Below:)
View horizon file
Convert DOS to Unix
Choose horizon type:
gridded (e.g. EarthVision)
Number of header lines to skip:
0
Total number of columns:
5
Column number of line_no:
1
Column number of cdp_no:
2
Column number of time or depth picks:
5
znull value (indicates missing pick):
-999999
Vertical axis of picked surface?
Positive Down
Vertical Units of Picked Horizons:
ms
(c) 2008-2015 AASPI - The University of Oklahoma
Execute kmeans3d
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

Horizon definition

The horizon definition panel will look the same for almost all AASPI GUIs:

1. Start time (upper boundary) of the analysis window.
2. End time (lower boundary of the analysis window.
3. Toggle that allows one to do the analysis between the top and bottom time slices described in 1 and 2 above, or alternatively between two imported horizons. If *USE HORIZON* is selected, all horizon related options will be enabled. If the horizons extend beyond the window limits defined in 1 and 2, the analysis window will be clipped.
4. Browse button to select the name of the upper (shallower) horizon.
5. Button that displays the horizon contents (see Figure 1).
6. Button to convert horizons from Windows to Linux format. If the files are generated from Windows based software (e.g. Petrel), they will have the annoying carriage return (^M) at the end of each line (Shown in Figure 1). Use these two buttons to delete those carriage returns. Note: This function depends on your Linux environment. If you do not have the program **dos2unix** it may not work. In these situations, the files may have been automatically converted to Linux and thus be properly read in.
7. Browse button to select the name of the lower (deeper) horizon.
8. Button that displays the horizon contents (see Figure 1).
9. Button to convert horizons from Windows to Linux format. (see 6 above).
10. Toggle that selects the horizon format. Currently *gridded* (e.g. EarthVision in Petrel) and *interpolated* (ASCII free format, e.g. SeisX) formats are supported. The gridded horizon are nodes of B-splines used in mapping and have no direct correlation to the seismic data survey. For example, gridded horizons may be computed simply from well tops. The x and y locations are aligned along north and east axes. In contrast interpolated horizons have are defined by *line_no, cdp_no (crossline_no)* and *time* triplets for each trace location. Examples of both format are shown in Figure 1. If *interpolated* is selected, the user needs to manually define each column in the file.
11. Number of header lines to skip in the *interpolated* horizon files.
12. Total number of columns in the *interpolated* horizon files.
13. Enter the column number containing the *line_no (inline_no)* of the interpolated data triplet.
14. Enter the column number containing the *cdp_no (crossline_no)* of the interpolated data triplet.
15. Enter the column number containing the *time* or *depth* value of the interpolated data triplet.
16. *Znull* value (indicate missing picks) in the horizon files.
17. Toggle to choose between positive down and negative down for the horizon files (e.g. Petrel uses negative down).
18. Choose the vertical units used to define the horizon files (either *s*, ms, *kft*, *ft*, *km*, or *m*).

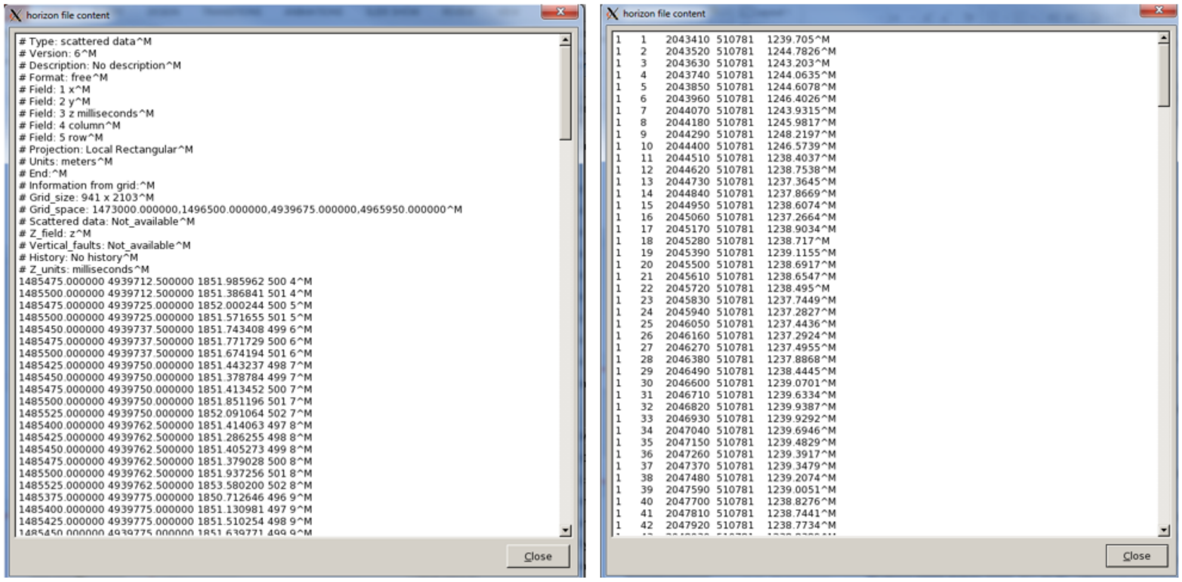
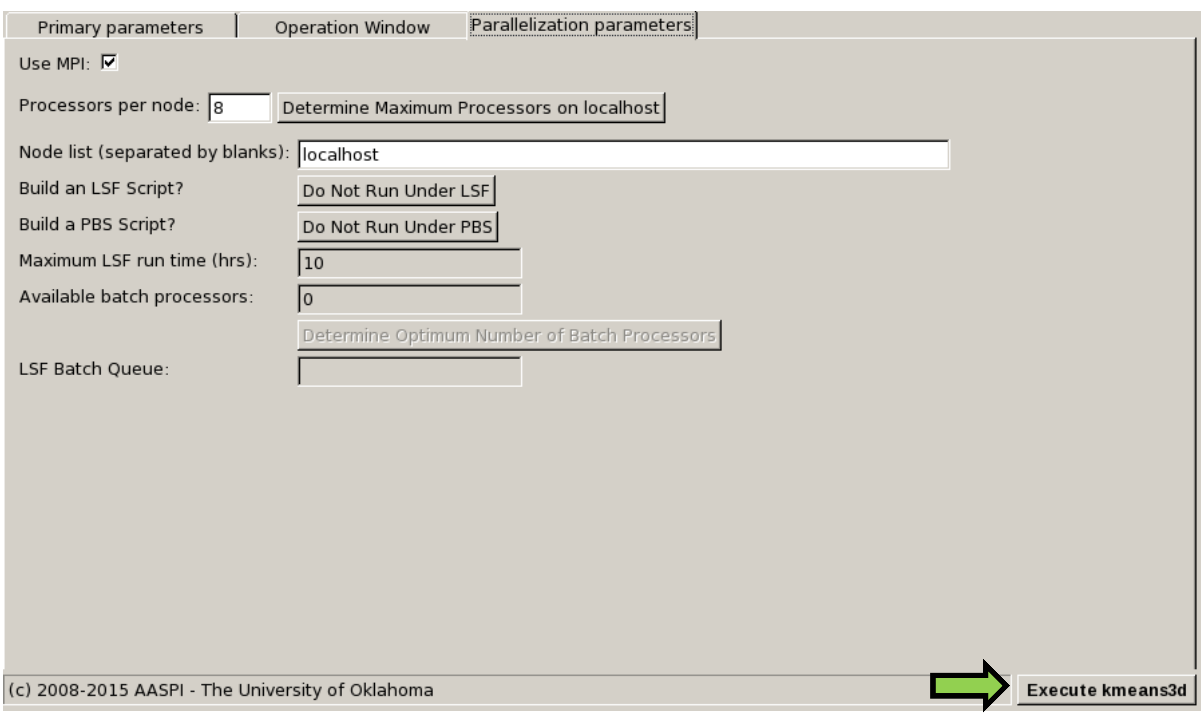


Figure 1. (left) A gridded horizon file (EarthVision format). (right) An interpolated horizon file with five columns (ASCII free format).

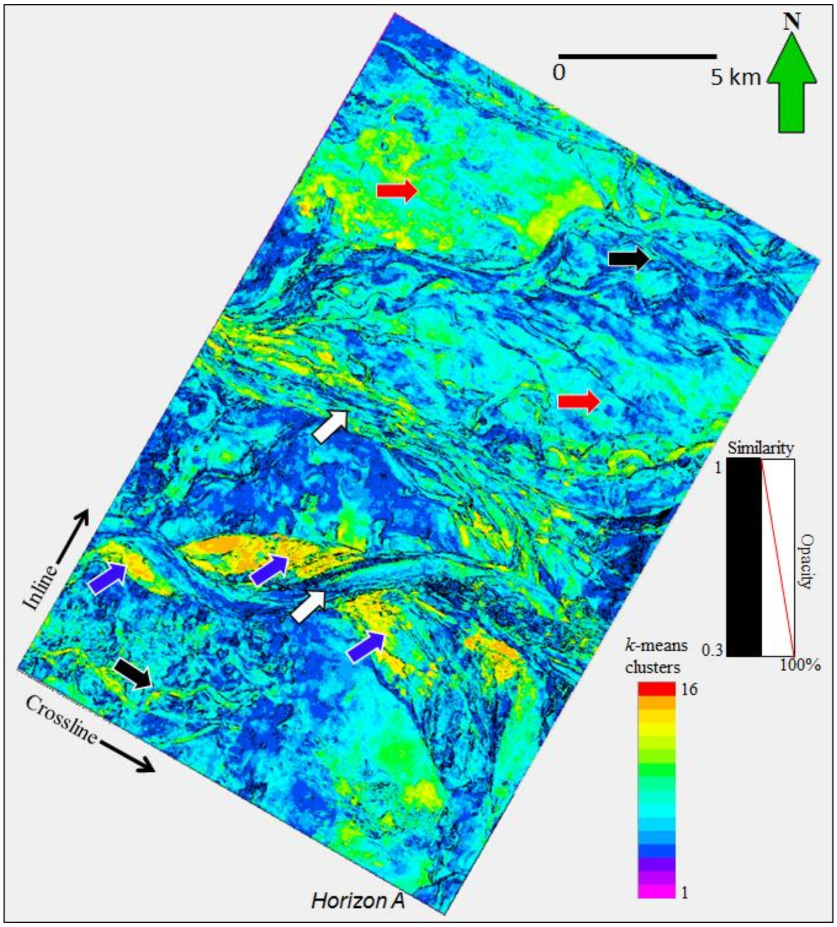
After defining the parallelization parameters, press the *Execute kmeans3d*.



The generated facies file is named as:
kmeans_cluster_number_${unique project name}_${suffix}.H

Visualization of the result

To view the resulted facies map, a user can either use aaspi_plot, or import the facies volume into another commercial seismic interpretation software. If using a commercial interpretation software, remember to use a discrete colorbar, or turn off the value interpolation, because the values in a *k*-means generated facies map are discrete values. The figure below is a facies map from *k*-means along a horizon in a turbidite system in Canterbury basin, New Zealand. We interpret the white arrows as multistoried channels, black arrows as sinuous channel complexes, blue arrows as a sand filled channel, and red arrows as slope fans.



References

MacQueen, J., 1967, Some methods for classification and analysis of multivariate observations: in L. M. Le Cam, and J. Neyman, eds., Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, 281–297.

Zhao, T., V. Jayaram, A. Roy, and K. J. Marfurt, 2015, A comparison of classification techniques for seismic facies recognition: Interpretation, **3**, SAE29-SAE58.